Comput. Math. Applic. Vol. 13, No. 7, pp. 635–646, 1987
Printed in Great Britain.

0097-4943/87 $3.00 + 0.00

AN EFFECTIVE DIFFERENCE SCHEME FOR QUASILINEAR DIFFUSION

G. S. GIPSON
School of Civil Engineering, Oklahoma State University, Stillwater, OK 74078, U.S.A.

(*Received* 3 *November* 1986)

Communicated by E. Y. Rodin

Abstract—Using classical finite difference techniques, a stable algorithm for the numerical solution of the quasilinear diffusion equation is developed using heuristic extensions of principles that apply rigorously to the linear case. The technique is proved to be suitable for the linear problem and is demonstrated to be viable for the quasilinear form of the equation. A stability criterion and accuracy assessment are made for the linear case. A quasilinear problem is solved using the method and its solution is compared with both the known analytical solution and a solution obtained by the classical Crank–Nicholson method. The results obtained by the improved algorithm are shown to correspond excellently with the exact solution, while the Crank–Nicholson approach becomes unstable. The possibility of a hybrid approach combining classical Crank–Nicholson and the improved algorithm is also discussed.

INTRODUCTION

One of the more useful governing equations that occurs in several branches of physics and engineering is the basic parabolic partial differential equation,

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(K\frac{\partial u}{\partial x}\right), \tag{1}$$

which will be denoted here by the generic term, diffusion equation. In this context, u is the concentration, K is the diffusivity and (x, t) are the independent spatial and temporal variables, respectively. Of course, other interpretations for u and K exist in many areas of science and engineering. Equation (1) has been exhaustively studied both analytically and numerically [1, 2], yet still presents a challenge in nearly all cases except for those in which K is functionally trivial. The finite difference formulas for this equation admit to rigorous analysis based primarily upon existing analytical solutions. Accordingly, the diffusion constant will be assumed to be constant for the stability and accuracy analysis presented in this paper. However, the technique will be applied to a quasilinear case of equation (1) by appealing to intuitive arguments and simply relying upon heuristic extensions of stability and accuracy principles derived for the linear case. The success of the algorithm when applied to a quaslinear problem will be demonstrated and the intuitive idea will be validified by comparing the numerical solution to a known analytical solution.

THE PROBLEM

We shall assume some standard but fairly general boundary conditions in order to properly pose the mathematical problem. In its entirety, the problem is to solve for $u(x, t)$ such that

$$\frac{\partial}{\partial x}\left(K\frac{\partial u}{\partial x}\right) = \frac{\partial u}{\partial t}, \quad 0 \leqslant x \leqslant \mathrm{L}; \quad 0 \leqslant t < \infty,$$

subject to

$$\begin{aligned} &u(x, 0) = f(x) \quad \text{for} \quad 0 \leqslant x \leqslant \mathrm{L} \\ &u(0, t) = u_0 \quad \text{for} \quad 0 \leqslant t \\ &u(\mathrm{L}, t) = 0 \quad \text{for} \quad 0 < t; \end{aligned} \tag{2}$$

L is a fixed constant.

In the traditional finite difference attack on this problem, a rectangular mesh is superimposed on the rectangular domain $\{0 \leqslant x \leqslant L;\ 0 \leqslant t\}$. The values of the function u at the mesh points are obtained from a knowledge of the initial and boundary conditions given by equation (2), and a suitable interpolation formula. Typically, the following approximations are made:

$$\frac{\partial u}{\partial t} = u_t \cong \frac{u(x, t + \Delta t) - u(x, t)}{\Delta t}$$

for a small, but finite value of Δt;

$$\frac{\partial u}{\partial x +} = u_{x+} \cong \frac{u(x + \Delta x, t) - u(x, t)}{\Delta x}$$

is the forward difference formula, for a small value Δx;

$$\frac{\partial u}{\partial x -} = u_{x-} \cong \frac{u(x, t) - U(x - \Delta x, t)}{\Delta x}$$

is the corresponding backward difference operator. It follows that the second derivatives are given by

$$u_{xx} \cong \frac{u_{x+} - u_{x-}}{\Delta x} \cong \frac{u(x + \Delta x, t) - 2u(x, t) + u(x - \Delta x, t)}{(\Delta x)^2}.$$

Referring to Fig. 1, we see the difference grid subdivided horizontally into increments of size $\Delta x = L/N$ and vertically into segments of length $\Delta t = T/M$. Each mesh point (j, k) is identified as P_j^k where $j = 0, 1, \ldots, N$ and $k = 0, 1, \ldots, M$. Denoting the value of the temperature at P_j^k as u_j^k, equation (1) has the discrete form

$$K\frac{u_{j+1}^k - 2u_j^k + u_{j-1}^k}{(\Delta x)^2} - \frac{u_j^{k+1} - u_j^k}{\Delta t} \cong 0. \tag{3}$$

Note that the diffusivity is assumed to be constant. Equation (3) has the significant feature of being very simple to visualize and program; from Fig. 1, it is readily seen that the values of u in the $k + 1$ row are given by the values of u in the k row.

The simplicity of the algorithm is marred by the fact that instability, the manifestation of error propagation, may occur. Defining $r = K\Delta t/(\Delta x)^2$, the interpolation formula (3) becomes

$$(u_{j+1}^k - 2u_j^k + u_{j-1}^k)r \cong u_j^{k+1} - u_j^k. \tag{4}$$

The practical implementation of the algorithm suggested by equation (4) involves judicious monitoring of the sensitivity of the solution to the value of r. To illustrate the problem, make the substitution

$$u_{j+q\Delta x}^{k+p\Delta t} = e^{\gamma(t+p\Delta t)}\, e^{i\beta(x+q\Delta x)}, \tag{5}$$

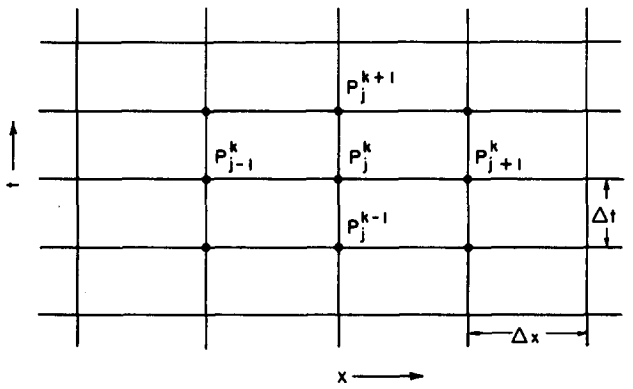

Fig. 1. The standard finite difference grid illustrating the subscript convention for the mesh points. The vertical dimension is taken over a large period T; the spacing is $\Delta t = T/M$. The N horizontal segments have length $\Delta x = L/N$.

where $i = \sqrt{-1}$ in equation (4). This leads quickly to the result

$$e^{\gamma\Delta t} = 1 - 4r \sin^2(\beta \Delta x/2). \tag{6}$$

Not only must the relationship between Δx and Δt expressed by equation (6) be satisfied, there is another implicit restriction that must be adhered to. Because $e^{\gamma t + i\beta x}$ is representative of the solution at P_j^k, the solution at P_j^{k+1} is $e^{\gamma(t+\Delta t)} e^{i\beta x}$. Thus, the factor $e^{\gamma\Delta t}$ must be chosen to have an absolute value <1 or else the approximate solution will diverge. The observation leads to the well-known *Von Neumann stability criterion*:

$$|1 - 4r \sin^2(\beta \Delta x/2)| \leqslant 1.$$

The inequality is satisfied iff $r \leqslant 1/2$, and herein lies one of the most objectionable features of practical finite difference analysis. That is, having chosen a convenient Δx, Δt is unduly restricted by the stability criterion on r.

Numerous schemes have been proposed for the purpose of avoiding this inherent instability (Fig. 2). Early attempts to circumvent the problem involved using a backward time difference; that is, approximating the derivative in the $k+1$ row instead of the k row. An elaborate scheme was suggested in the classic reference by Crank and Nicholson [3] and that algorithm is still the most popular for the diffusion equation. In this method, the derivative was approximated by averaging the spatial differences in the kth and $k+1$ rows. These methods entail mathematical difficulties not present in the algorithm suggested by equation (3). The latter is explicit in the single unknown quantity u_j^{k+1} whereas implicit methods characterized by Crank–Nicholson involve several unknowns. As a result, a system of equations must be solved at each time step. Furthermore, if the differential equation is nonlinear, the resulting system of algebraic equations obtained from the Crank–Nicholson analog is also nonlinear, requiring tedious iteration methods for its solution.

Massive amounts of research have been conducted in order to refine and replace the Crank–Nicholson method (a brief survey may be found in Chap. 4 of Ref. [2]). The method proposed in the next section may be interpreted as such a refinement. Expressions are obtained which relate analytically the weight functions to the mesh size, yielding a "built-in" higher order of accuracy. The proposed analog is exceptionally simple to program, even in a nonlinear case. This consideration is important because systems programmers implementing finite differences alone, or finite differences coupled with the more modern finite element [4] or boundary element techniques [5], are in constant need of better algorithms.

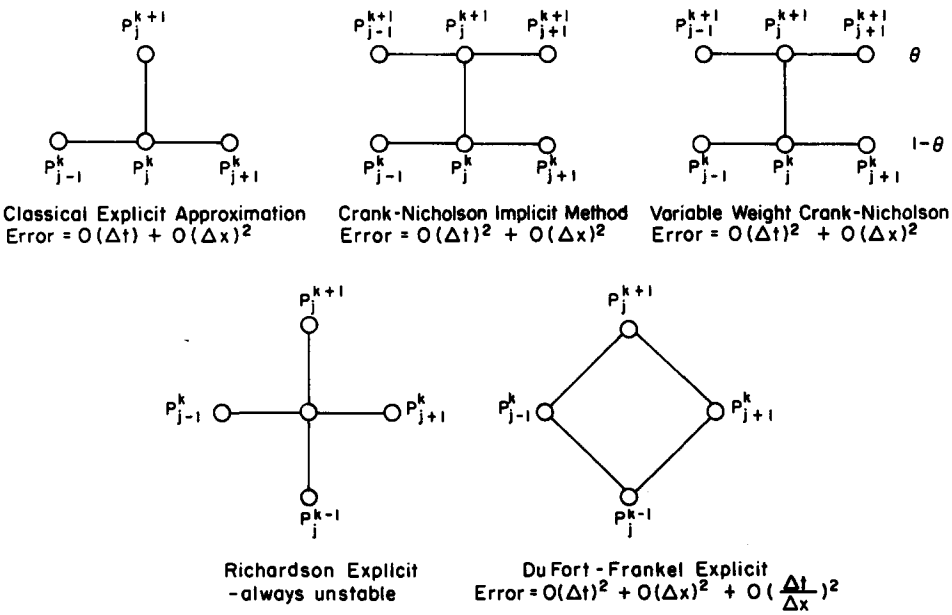

Fig. 2. Several common finite differences stencils for solving the heat transfer equation.

DEVELOPMENT OF THE ALGORITHM

The formula proposed here is distinctive by virtue of the manner in which the time derivative is approximated. The space derivative involves averaging the k and $k+1$ rows, but the approximation of the time derivation is achieved by averaging the time derivatives of three columns, as illustrated by Fig. 3. Readers familiar with the work of Richtmeyer and Morton [6] will recognize this difference molecule as being similar to scheme 12 on p. 191 of that work. We will now show that this is indeed the case and, in the next section, will further show the validity of the formula for analyzing the quasilinear equation. Denote by θ and α weighting values such that

$$\alpha = (1-\theta)/2. \tag{7}$$

With this, the proposed difference analog is

$$\alpha(u_{j+1}^{k+1} - u_{j+1}^{k}) + \theta(u_j^{k+1} - u_j^k) + \alpha(u_{j-1}^{k+1} - u_{j-1}^k) = (u_{j+1}^k - 2u_j^k + u_{j-1}^k)r, \tag{8}$$

where $r = K\Delta t/(\Delta x)^2$ as before. Assume that

$$u_j^k = X(x_j)T(t_k) \tag{9}$$

as a standard separation of variables effort. Substituting equation (9) into equation (8) yields

$$\frac{T_{k+1} - T_k}{T_k} = \frac{r(X_{j+1} - 2X_j + X_{j-1})}{\alpha(X_{j+1} + X_{j-1}) + \theta X_j} = -\mathrm{C}, \tag{10}$$

where C is a positive constant signifying the separation. The temporal portion of equation (10) is solved by

$$T_k = (1-\mathrm{C})^k. \tag{11}$$

The spatial equation, after simplification, is

$$X_{j+1} + [(2\mathrm{C}\theta - 4r)/(2r + \mathrm{C} - \mathrm{C}\theta)]X_j + X_{j-1} = 0. \tag{12}$$

It is easy to show that a solution of equation (12) is

$$X_j = \sin j\beta, \tag{13}$$

provided that

$$\cos\beta = (2r - \mathrm{C}\theta)/[2r + \mathrm{C}(1-\theta)]. \tag{14}$$

In order to obtain a solution to a specific problem, consider an insulated rod with a two-unit length with an initially symmetrically centered temperature distribution $f(x)$, and with ends held at zero temperature. In this case, the boundary conditions are

$$u(x,0) = f(x) \quad \text{for} \quad 0 \leqslant x \leqslant 1, \tag{15}$$

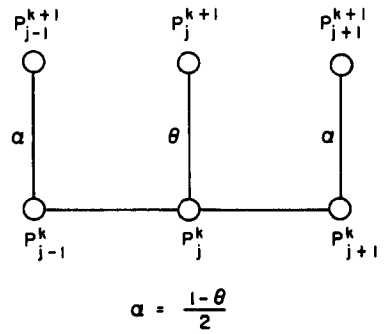

Fig. 3. The proposed algorithm which uses column averaging for success. The molecule is substantially the same as scheme 12 presented by Richtmeyer and Morton [6, p. 191]. Error $= O(\Delta t)^2 + O(\Delta x)^4$.

$$u(0, t) = 0 \quad \text{for} \quad 0 < t \tag{16}$$

and

$$\frac{\partial}{\partial x}[u(1, t)] = 0 \quad \text{for} \quad 0 \leqslant t. \tag{17}$$

It is clear that $\sin j\beta$ satisfies $u_0^k = 0$ if $j = 0$. Letting $h\Delta x = 1$, symmetry provides that $u_{m+1}^k = u_{m-1}^k$ or equivalently $\sin(h + 1)\beta = \sin(h - 1)\beta$. Elementary trigonometric identities convert this to the requirement that $\sin\beta \cos h\beta = 0$. Therefore, for a nontrivial result, we must have

$$h\beta = \pi/2, 3\pi/2, \ldots, (2h - 1)\pi/2. \tag{18}$$

Using the fact that the solution u_j^k is a sum of products of terms like those in equations (11) and (13), we get

$$u_j^k = \sum_{\ell=1}^{h} [(1 - C)^k] \sin\frac{j(2\ell - 1)}{2h}\pi. \tag{19}$$

The constant C is given by equation (14) as

$$C = 2r(1 - \cos\beta)/[(1 - \theta)\cos\beta + \theta]. \tag{20}$$

Recalling that C is positive and noting that the numerator of equation (20) is always nonnegative, we must ensure that the denominator is positive with the requirement

$$0 > \cos\beta/(\cos\beta - 1). \tag{21}$$

Making the obvious substitutions yields an explicit series solution to the problem. Of primary interest here is the term bracketed in equation (19), the so-called *growth factor*. It is a geometric progression in k. Hence, we have for $0 < C < 1$ an unconditionally decaying progression; if $1 < C < 2$, equation (20) indicates that the sign of $(1 - C)^k$ alternates and the progression decays. Outside the region $0 < C < 2$, the progression becomes unbounded. Therefore, the stability requirement is

$$0 < r(1 - \cos\beta)/[\theta + (1 - \theta)\cos\beta] < 1$$

or, equivalently

$$0 < r < [\theta(1 - \cos\beta) + \cos\beta]/(1 - \cos\beta). \tag{22}$$

Because $\cos\beta/(\cos\beta - 1) > 1) > 0.5$, the stability criterion equation (22) reduces to

$$0 < r < \theta - 0.5. \tag{23}$$

The accuracy of the formulation is dictated by the difference between the value of the differential equation and the value of the formula at P_j^k. This quantity is commonly called the *truncation error*. In order to analyze it, we will need the Taylor's series expansions

$$u_{j+1}^k \cong u(x + \Delta x, t) = u(x, t) + u'\Delta x + \frac{u''(\Delta x)^2}{2!} + \cdots,$$

where the primes denote spatial differentiation; and

$$u_j^{k+1} \cong u(x, t + \Delta t) = u(x, t) + u\Delta t + u\frac{(\Delta t)^2}{2!} + \cdots,$$

where the dots indicate temporal differentiation. Similar expansions hold for u_{j+1}^{k+1}, u_{j-1}^{k+1} etc. Denoting the truncation error by $T(u)$, we have by definition

$$T(u)\Delta t = \alpha(u_{j+1}^{k+1} - u_{j+1}^k) + \theta(u_j^{k+1} - u_j^k) + \alpha(u_{j-1}^{k+1} - u_{j-1}^k) - r(u_{j+1}^k - 2u_j^k + u_{j-1}^k) - \Delta t\left\{\frac{\partial u}{\partial t} - K\frac{\partial^2 u}{\partial x^2}\right\}_j^k.$$

The quantity in curly brackets is identically zero by hypothesis. Thus,

$$T(u)\Delta t = \theta u_j^{k+1} + (2r - \theta)u_j^k - (\alpha + r)u_{j+1}^k - (\alpha + r)u_{j-1}^k + \alpha u_{j+1}^{k+1} + \alpha u_{j-1}^{k+1}.$$

Substituting the Taylor's series analogs into the discrete quantities, we obtain

$$\begin{aligned}T(u)\Delta t = {} & \theta[u + \dot{u}\Delta t + \ddot{u}(\Delta t)^2/2! + \dddot{u}(\Delta t)^3/3! + \cdots] \\ & + 2r - \theta u - (\alpha + r)[u + u'\Delta x + u''(\Delta x)^2/2! \\ & + u'''(\Delta x)^3/3! + \cdots] - (\alpha + r)[u - u'\Delta x + u''(\Delta x)^2/2! \\ & - u'''(\Delta x)^3/3! + \cdots] + \{u + u'\Delta x + \dot{u}\Delta t + [u''(\Delta x)^2 \\ & + 2\dot{u}'\Delta x\Delta t + \ddot{u}(\Delta t)^2]/2! + [u'''(\Delta x)^3 + 3\dot{u}''(\Delta x)^2\Delta t \\ & + 3\ddot{u}'(\Delta x)(\Delta t)^2 + \dddot{u}(\Delta t)^3]/3! + [u^{iv}(\Delta x)^4 + \cdots]/4! \\ & + \cdots\} + \{u - u'\Delta x + \dot{u}\Delta t + [u''(\Delta x)^2 - 2\dot{u}'\Delta x\Delta t \\ & + \ddot{u}(\Delta t)^2]/2! + [-u'''(\Delta x)^3 + 3\dot{u}''(\Delta x)^2\Delta t - 3\ddot{u}'\Delta x(\Delta t)^2 \\ & + \dddot{u}(\Delta t)^3]/3! + [u^{iv}(\Delta x)^4 + \cdots]/4! + \cdots\}. \end{aligned} \tag{24}$$

Noting that

$$K\frac{\partial^4 u}{\partial x^4} = \frac{\partial^3 u}{\partial t \partial x^2}$$

and that

$$K\frac{\partial^3 u}{\partial t \partial x^2} = \frac{\partial^2 u}{\partial t^2}$$

we get

$$K^2\frac{\partial^4 u}{\partial x^4} = \frac{\partial^2 u}{\partial t^2}.$$

Also, it follows directly from the definition of r that $K^2(\Delta t)^2 = r^2(\Delta x)^4$. With these identities, equation (24) reduces to

$$T(u)\Delta t = u^{iv}(\Delta x)^4[r^2\theta/2 - r/12 + r^2(1-\theta)/2 + r(1-\theta)/2] + \theta u(\Delta t)^3/6 - ru^{vi}(\Delta x)^6/360 + \cdots,$$

giving

$$T(u) = u^{iv}(\Delta x)^2[(6r - 6\theta + 5)/12] + \theta u(\Delta t)^2/6 - u^{vi}(\Delta x)^4/360 + \cdots. \tag{25}$$

Equation (25) states that for $r = \theta - 5/6$, the truncation error is of the order $O(\Delta t)^2 + O(\Delta x)^4$. Note that $\theta = r + 5/6$ insures stability in accordance with equation (23).

As predicted, with this stability criterion invoked, the algorithm defined by equation (8) becomes identical to that presented by Richtmeyer and Morton [6, p. 191].

EXTENSION TO THE NONLINEAR CASE

Although the difference equation (8) was analyzed for the linear case, it was with the quasilinear equation in mind that this was done. Thus, from a heuristic standpoint, equation (8) has been contrived to be ideally suited algorithmically for the solution of equations of the type

$$\frac{\partial}{\partial x}\left[K(x, t, u)\frac{\partial u}{\partial x}\right] = \frac{\partial u}{\partial t}. \tag{26}$$

In this section, we will report on the solution of a case of equation (26), in which $K = K(u)$, using the presented difference scheme and compare the result with known analytical and numerical solutions as given by Richtmeyer and Morton [6] for a specific problem.

The model equation is the quasilinear form with $K = 5u^4$ [6]:

$$\frac{\partial}{\partial x}\left(5u^4\frac{\partial u}{\partial x}\right) = \frac{\partial u}{\partial t}; \tag{27}$$

the solution of equation (27) is given implicitly by

$$5u_0^4 \ln|u-u_0| + \tfrac{5}{4}(u-u_0)^4 + \tfrac{20}{3}u_0(u-u_0)^3 + 15u_0^2(u-u_0)^2 + 20u_0^3(u-u_0) = v_0(v_0 t - x + x_0), \quad (28)$$

where u_0, v_0 and x_0 are constants. Consistent with Richtmeyer and Morton [6], we perform the analysis over the unit interval $0 \leqslant x \leqslant 1$ and let $u_0 = 1$, $v_0 = 1500$ and $x_0 = 0.035$. The finite difference equation to be solved has the form

$$\alpha_j(u_{j+1}^{k+1} - u_{j+1}^k) + \theta_j(u_j^{k+1} - u_j^k) + \alpha_j(u_{j-1}^{k+1} - u_{j-1}^k) = [(u^5)_{j+1}^k - 2(u^5)_j^k + (u^5)_{j-1}^k]r, \quad (29)$$

where

$$r = \Delta t/(\Delta x)^2, \quad (30)$$

$$\theta_j = 5(u^4)_j^k r + 5/6 \quad (31)$$

and

$$\alpha_j = (1-\theta_j)/2. \quad (32)$$

In order to obtain the initial data for the difference algorithm, equation (28) was solved using the Newton–Raphson method at time $t = 0$. At each time step, equation (28) was again solved and the u_0^k and u_1^k computed were used as boundary values for the difference scheme. Since these values will be known, equation (29) may be written in the suggestive form

$$\alpha_j u_{j+1}^{k+1} + \theta_j u_j^{k+1} + \alpha_j u_{j-1}^{k+1} = [(u^5)_{j+1}^k - 2(u^5)_j^k + (u^5)_{j-1}^k]r + \theta_j u_j^k + \alpha_j[u_{j+1}^k + u_{j-1}^k]. \quad (33)$$

Because the r.h.s. of equation (33) is completely determined by known values of the variable at any time step, we may recast it as

$$\alpha_j u_{j+1} + \theta_j u_j + \alpha_j u_{j-1} = D_j, \quad (34)$$

where the superscript $k+1$ has been suppressed.

Now suppose existence of two sets E_j and F_j such that

$$u_j = E_j u_{j+1} + F_j, \quad (35)$$

in order to derive a recursion formula for the unknown u_j. Making the substitution $u_{j-1} = E_{j-1}u_j + F_{j-1}$ in equation (34) and solving for u_j yields

$$u_j = [(D_j - \alpha_j F_{j-1})/(\theta_j + \alpha_j E_{j-1})] - [\alpha_j/(\theta_j + \alpha_j E_{j-1})]u_{j+1}. \quad (36)$$

Equating expressions (34) and (36), we see that

$$E_j = -\alpha_j/(\theta_j + \alpha_j E_{j-1}) \quad (37)$$

and

$$F_j = (D_j - \alpha_j F_{j-1})/(\alpha_j + \alpha_j E_{j-1}). \quad (38)$$

Because equation (35) has a one-parameter family of solutions, $E_1 = 0$ and $F_1 = u_1$ is a solution; these values coupled with equations (37) and (38) define the sets E_j and F_j. Finally, the value of $u_{j=L}$ is given at the right-hand boundary, and thus equation (35) defines the u_j inductively in order of decreasing j.

The simplicity of the foregoing analysis is remarkable when compared with the parallel procedure applied to a typical modified Crank–Nicholson analog; i.e.

$$u_j^{k+1} - u_j^k = r\{\theta[(u^5)_{j+1}^{k+1} - 2(u^5)_j^{k+1} + (u^5)_{j-1}^{k+1}] + (1-\theta)[(u^5)_{j+1}^k - 2(u^5)_j^k + (u^5)_{j-1}^k]\},$$

where $0 \leqslant \theta \leqslant 1$. Proceeding as before, we separate the known and unknown quantities:

$$u_j^{k+1} - r\theta[(u^5)_{j+1}^{k+1} - 2(u^5)_j^{k+1} + (u^5)_{j-1}^{k+1}] = u_j^k + (1-\theta)[(u^5)_{j+1}^k - 2(u^5)_j^k + (u^5)_{j-1}^k]r. \quad (39)$$

The left member is nonlinear and the attempted analysis breaks down. Usually, an approximation is resorted to; typically,

$$(u^5)_j^{k+1} \cong (u^5)_j^k + 5(u^4)_j^k(u_j^{k+1} - u_j^k) \quad (40)$$

might be tried in an attempt to linearize the l.h.s. of equation (39). Substituting equation (40) into equation (39) yields an equation in the form

$$A_j u_{j+1}^{k+1} + B_j u_j^{k+1} + C_j u_{j-1}^{k+1} = D_j. \tag{41}$$

It is obvious that the coefficients in equation (41) will be rather abstruse mathematical entities, as opposed to the simple weight functions presented in the previous analysis.

RESULTS AND CONCLUSIONS

In order to demonstrate the superior capabilities of the derived algorithm, the analytical solution to equation (28), the Crank–Nicholson difference scheme, equation (39), and the "new" scheme

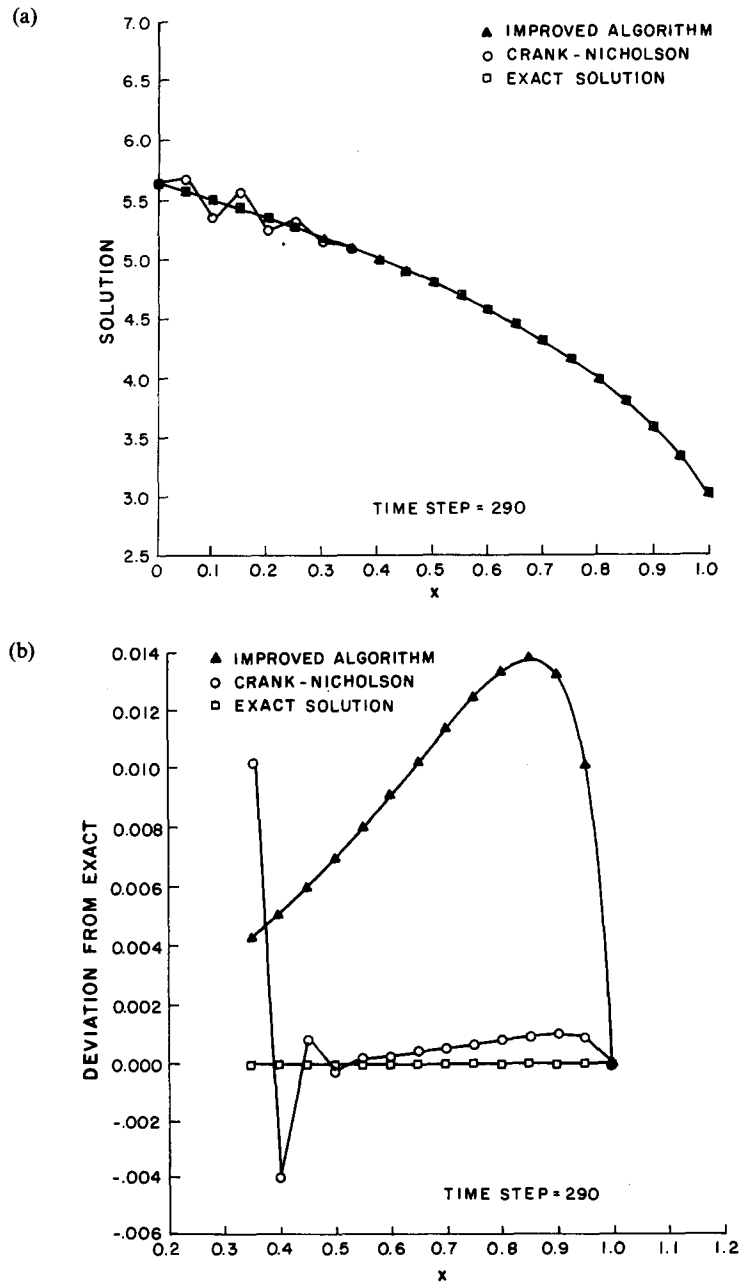

Fig. 4(a). The development of the solution as the instability develops in the classic Crank–Nicholson algorithm. Both the Crank–Nicholson and the improved algorithm are implemented with step sizes of $\Delta t/(\Delta x)^2 = 0.001$. (b). The derivation in the region past the worst instability. Note that the overall quality of Crank–Nicholson for $x \geqslant 0.45$ is much beter than that for the improved algorithm.

were programmed and the results were compared. Figs 4–6 show the results of one such test at three successive intervals in time where the classical Crank–Nicholson formula had demonstrated problems with instability [6]. For these runs, $\Delta t/(\Delta x)^2 = 0.001$. The two difference algorithms were programmed with identical parameters except that the relation $\theta = r + 5/6$ was used to insure the stability of the scheme derived in the previous section.

The Crank–Nicholson instability, which actually showed up visually near time step 286 (relative to the scale shown in the figures), is manifestly apparent at time step 290. This is illustrated in Fig. 4(a), and the propagation of the disturbance through time steps 300 [Fig. 5(a)] and 310 (Fig. 6) represents a classic example of the nature of finite difference instability. In contrast, the improved algorithm is seen to replicate the exact solution to a very high degree of correlation.

In order to accurately gauge the overall quality and magnitude of the deviations, Figs 4(b) and 5(b) are presented to illustrate the absolute deviations of the two difference solutions from the exact

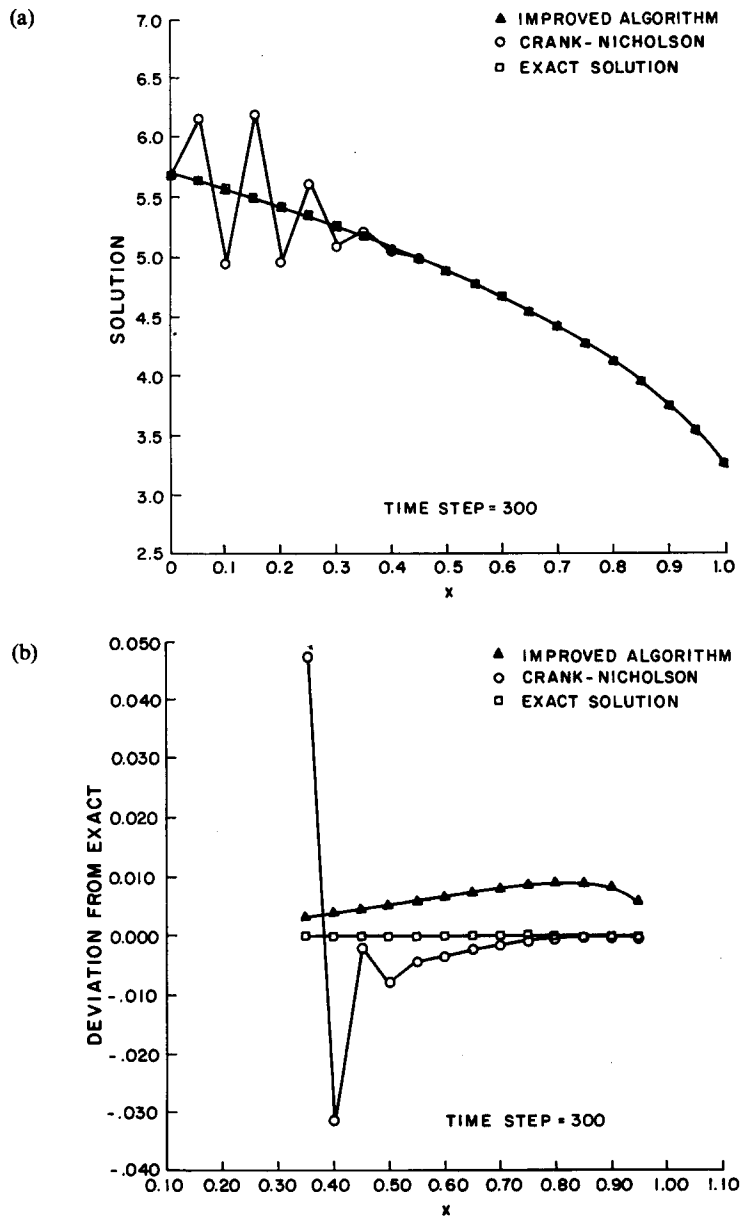

Fig. 5(a). The instability in the classic Crank–Nicholson becomes worse while the new algorithm is manifestly stable. (b). As in Fig. 4(b), the Crank–Nicholson algorithm demonstrates remarkable accuracy outside the unstable region.

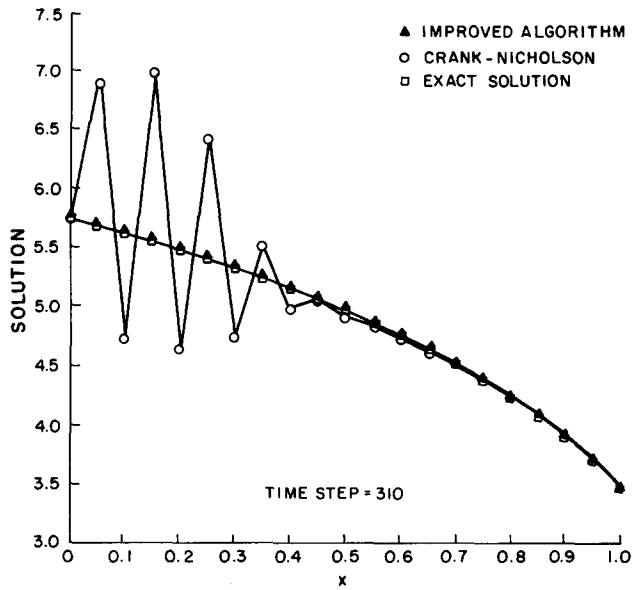

Fig. 6. The improved algorithm performs excellently as the Crank–Nicholson scheme develops oscillations of ever-increasing amplitude.

answers at time steps 290 and 300, respectively. These figures point out an interesting quirk in the improved algorithm in that it does not actually "improve" the results at all mesh points. As shown in Fig. 4(b), the classic Crank–Nicholson scheme demonstrates a much better prediction of the exact results outside the region of instability. This peculiar phenomenon occurred in fact throughout the solution at nearly all time steps. For instance, Fig. 7 shows a comparison of the deviations of the two difference algorithms from the exact answer at time step 60, well before the onset of the instability in Crank–Nicholson. Only the range $0 \leqslant x \leqslant 0.2$ is shown in Fig. 7 because the solution had not yet progressed to the stage where the results within $0.2 \leqslant x \leqslant 1.0$ had changed from the initial value of unity. On a percentage basis, the relative performance of the improved algorithm in comparison to the Crank–Nicholson solution is very poor. On an absolute basis, both methods give extremely good results.

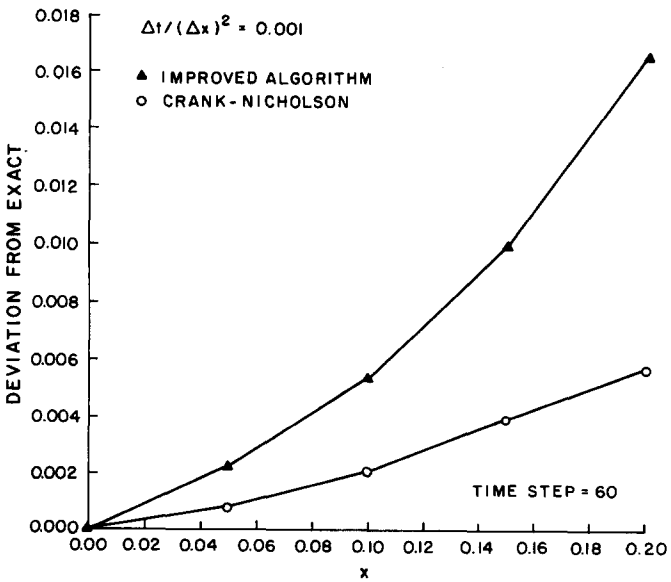

Fig. 7. The overall quality of the deviations in the Crank–Nicholson scheme and the new algorithm is demonstrated at a typical time step well before the problems with instability occur.

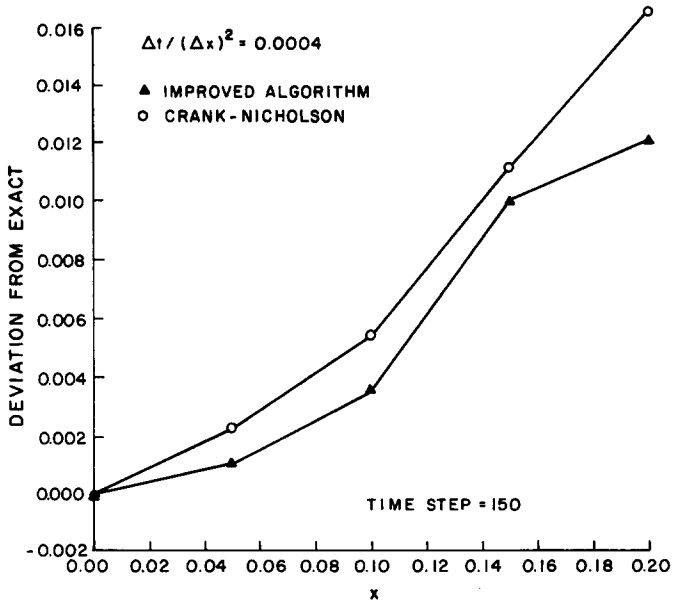

Fig. 8. The same solution deviation plot in time as in Fig. 7, computed with a stepping increment 2.5 times smaller. The improved algorithm is now the overall better scheme.

In most engineering work, especially of the type that motivated this study, the slightly less accurate global solution is hardly significant. The fact that the improved scheme churns its way through the region of Crank–Nicholson instability is of much more concern and vastly outweighs any accuracy argument that can be put forth in favor of Crank–Nicholson. Still, the fact that the new scheme consistently gave worse answers than Crank–Nicholson for the $\Delta t/(\Delta x)^2 = 0.001$ step size was troublesome, and some investigation into this phenomenon was warranted. It was found that the new algorithm was reprieved when the mesh size was reduced, and it was noticed that the overall improvement in quality of the new scheme was considerably more pronounced than that observed in the Crank–Nicholson algorithm. This result is confirmed in Fig. 8, wherein a comparison is made at time step 150 for $\Delta t/(\Delta x)^2 = 0.0004$. This is equivalent to the time step of 60 shown in Fig. 7. The improved algorithm is the clearly superior method from an accuracy standpoint for this step size. Although no effort was made to quantify the reasons why this was true, a sufficient number of numerical experiments were run to see that the new method's trend of ever-increasing accuracy and superiority for decreasing step size was unmistakable. Therefore, we must conclude that the improved algorithm is the overall superior technique for this type of problem.

However, the previous discussion leads to a further interesting conjectural conclusion. Since the Crank–Nicholson scheme and the improved method are effectively comparable in programming difficulty and computer usage time, a hybrid approach may be the most advantageous method of attack in a problem of the type defined by equation (26). That is, classic Crank–Nicholson could be used with a relatively coarse mesh until the onset of instability; at that time, the improved algorithm could be implemented. Since most modern finite difference codes are designed to constantly monitor stability, it would appear to be a simple matter to write an adaptive program to perform this analysis.

Finally, we note that the entire preceding analyses were performed for one space variable. However, the programming of the algorithm for more than one spatial dimension should be a relatively straightforward task. It would be expected that the tendency of the new method toward superior behavior with respect to stability would continue in the multidimensional case as well. Also, the new method should be viewed as a viable contender in a coupled finite difference–finite element or finite difference–boundary element system in which the time stepping is performed with the difference scheme.

Acknowledgements—The author would like to thank Ms Susan Sartwell and Ms Charlene Fries for their careful preparation of the manuscript, and Mr J. C. Ortiz for help in preparing the figures.

REFERENCES

1. W. F. Ames, *Nonlinear Partial Differential Equations in Engineering*. Academic Press, New York (1965).
2. L. Lapidus and G. F. Pinder, *Numerical Solution of Partial Differential Equations in Science and Engineering*. Wiley, New York (1982).
3. J. Crank and P. Nicholson, A practical method for numerical evaluation of partial differential equations of the heat conduction type. *Proceedings of the Cambridge Philosophical Society* **43**, 50 (1947).
4. O. C. Zienkiewicz, *The Finite Element Method*, 3rd edn. McGraw-Hill, London (1978).
5. P. K. Banerjee and R. Butterfield, *Boundary Element Methods in Engineering Science*. McGraw-Hill, London (1981).
6. R. C. Richtmeyer and K. W. Morton, *Difference Methods for Initial Value Problems*, 2nd edn. Interscience, New York (1967).